```
function
[BestResults,tabela_Pg,tabela_Vg,tabela_T,tabela_Qc,Vviolat,Qgviolat,ta
bela_cvorovi,tabela_grane,Statistics]=st(L,testsistem,Case,N,max_it,BRP
,FBEST,TOC);

% =============================================

if nargin < 3
    Case = 'Fcost'; N=25;max_it=100;BRP=1;FBEST=1;TOC=1;
end

switch testsistem
      case 'Caso_9Bus'

[mreza,cvorovi,generatori,transformatori,kompenzatori,Vpqmin,Vpqmax,Sba
se]=Caso_9Bus;
      case 'Caso_30Bus'

[mreza,cvorovi,generatori,transformatori,kompenzatori,Vpqmin,Vpqmax,Sba
se]=Caso_30Bus;
end
%----------------------------------------------------------------------
----
[m,n]=size(cvorovi);cvor=cvorovi(:,1);tip=cvorovi(:,2);
V0=cvorovi(:,3);teta0=cvorovi(:,4);Pg=cvorovi(:,5);Qg=cvorovi(:,6);
ppot=cvorovi(:,7);qpot=cvorovi(:,8);Vbase=cvorovi(:,10);
%----------------------------------------------------------------------
----
t=mreza(:,7);
%----------------------------------------------------------------------
----
[ngen,nkolg]=size(generatori);gcvor=generatori(:,1);Vgmin=generatori(:,
2);
Vgmax=generatori(:,3);Pgmin=generatori(:,4);Pgmax=generatori(:,5);
Qgmin=generatori(:,6);Qgmax=generatori(:,7);
a=generatori(:,8);b=generatori(:,9);c=generatori(:,10);d=generatori(:,1
1);e=generatori(:,12);
%----------------------------------------------------------------------
----
[ntf,nkolt]=size(transformatori);redbrt=transformatori(:,1);
tmin=transformatori(:,2);tmax=transformatori(:,3);tfbusi=transformatori
(:,4);tfbusj=transformatori(:,5);
%----------------------------------------------------------------------
----
[nkom,nkolk]=size(kompenzatori);kcvor=kompenzatori(:,1);
Qgcmin=kompenzatori(:,2);Qgcmax=kompenzatori(:,3);
%----------------------------------------------------------------------
----
```

```
npq=0; npv=0;
for k=1:m
    if tip(k)==0
        sl=cvor(k);
    elseif tip(k)==2
        npq=npq+1;
        pq(npq)=cvor(k);
    elseif tip(k)==1
        npv=npv+1;
        pv(npv)=cvor(k);
    end
end
%-----------------------------------------------------------------------
----
for k=1:npv
Pg(pv(k))=L(k);
end
for k=(npv+1):(npv+ngen)
V0(gcvor(k-npv))=L(k);
end
for k=(npv+ngen+1):(npv+ngen+ntf)
t(redbrt(k-npv-ngen))=L(k);
end
for k=(npv+ngen+ntf+1):(npv+ngen+ntf+nkom)
Qg(kcvor(k-npv-ngen-ntf))=L(k);
end
%-----------------------------------------------------------------------
----
pinj=Pg-ppot;
qinj=Qg-qpot;
%-----------------------------------------------------------------------
----
[Y,yg,yg0]=Ybus(mreza,cvorovi,t);
G=real(Y); B=imag(Y);
ggr=real(yg); bgr=imag(yg);
ggr0=real(yg0); bgr0=imag(yg0);
B1=B-bgr0;
for k=1:m
    if tip(k)==0
        B1(:,k)=[];
        B1(k,:)=[];
    end
end
B2rr=B-bgr0;
for k=1:m
    if tip(k)==0 || tip(k)==1
        B2rr(:,k)=0;
        B2rr(k,:)=0;
    end
    if tip(k)==2
        index(k)=k;
    end
end
index=nonzeros(index);
for k=1:length(index)
    B2r(k,:)=B2rr(index(k),:);
end
for k=1:length(index)
    B2(:,k)=B2r(:,index(k));
end
%----------------------------------------------------------------------
----
iter=0; itermax=100; epsilon=0.00001; maxraz=epsilon+1;
V=V0;teta=teta0; myvars=struct;
while (iter < itermax) && (maxraz > epsilon)
iter=iter+1;
tetanew=teta;Vnew=V;
VolBus(:,iter)=V;
save test.mat VolBus iter;
    for k=1:m
        suma=0;
        for j=1:m
            suma=suma+Vnew(j)*(G(k,j)*cos(tetanew(k)-tetanew(j))...
                           +B(k,j)*sin(tetanew(k)-tetanew(j)));
        end
            deltaP(k)=pinj(k)-Vnew(k)*suma;
    end
    deltaP(sl)=[];
    Vnew(sl)=[];
    tetanew(sl)=[];
    teta(sl)=[];
    dteta=-inv(B1)*(deltaP'./Vnew);
    teta=tetanew+dteta;
    tetapomoc=zeros(m,1);
    tetapomoc(sl)=teta0(sl);
    tetapomoc(1:sl-1)=teta(1:sl-1);
    tetapomoc(sl+1:m)=teta(sl:m-1);
    teta=tetapomoc;
    Vnewpomoc=zeros(m,1);
    Vnewpomoc(sl)=V0(sl);
    Vnewpomoc(1:sl-1)=Vnew(1:sl-1);
    Vnewpomoc(sl+1:m)=Vnew(sl:m-1);
    Vnew=Vnewpomoc;
    for k=1:m
        suma=0;
        for j=1:m
```

```
            suma=suma+Vnew(j)*(G(k,j)*sin(teta(k)-teta(j))...
                           -B(k,j)*cos(teta(k)-teta(j)));
        end
        deltaQ(k)=qinj(k)-Vnew(k)*suma;
    end
B2i=-inv(B2);
br1=0;
for k=1:m
    if tip(k)==2
        br1=br1+1;
        Vpq(br1)=Vnew(k);
        deltaQpq(br1)=deltaQ(k);
    end
end
for kk=1:npq
deltaV(kk)=B2i(kk,:)*(deltaQpq'./Vpq');
Vpqnew(kk)=Vpq(kk)+B2i(kk,:)*(deltaQpq'./Vpq');
end
br=0;
for k=1:m
    if tip(k)==2
    br=br+1;
    V(k)=Vpqnew(br);
    end
end

maxraz=max(deltaP);
end
%-----------------------------------------------------------------------
----
for k=1:m
    sumap=0;
    sumaq=0;
    for l=1:m
        sumap=sumap+V(l)*(G(k,l)*cos(teta(k)-teta(l))...
                         +B(k,l)*sin(teta(k)-teta(l)));
        sumaq=sumaq+V(l)*(G(k,l)*sin(teta(k)-teta(l))...
                         -B(k,l)*cos(teta(k)-teta(l)));
    end
    pinj(k)=V(k)*sumap;
    qinj(k)=V(k)*sumaq;
    sinj(k)=pinj(k)+qinj(k)*sqrt(-1);
end

%-----------------------------------------------------------------------
----
%Pg=pgen;
Pg(sl)=pinj(sl)+ppot(sl);
```

```
Qg=qinj+qpot;
%----------------------------------------------------------------------
----
for i=1:m
for j=1:m
tokovip(i,j)=(V(i)^2)*ggr(i,j)-V(i)*V(j)*(ggr(i,j)*cos(teta(i)-teta(j))
...

+bgr(i,j)*sin(teta(i)-teta(j)));
tokoviq(i,j)=(-V(i)^2)*bgr(i,j)-(V(i)^2)*bgr0(i,j)-V(i)*V(j)...

*(ggr(i,j)*sin(teta(i)-teta(j))-bgr(i,j)*cos(teta(i)-teta(j)));
tokoviqX(i,j)=(-V(i)^2)*bgr(i,j)-V(i)*V(j)...

*(ggr(i,j)*sin(teta(i)-teta(j))-bgr(i,j)*cos(teta(i)-teta(j)));
end
end
br=1;
for ii=1:m
for jj=1:m
if abs(tokovip(ii,jj)+sqrt(-1)*tokoviq(ii,jj))~=0
tokovis(br,1:2)=[ii jj];
tokovis(br,3)=tokovip(ii,jj)+sqrt(-1)*tokoviq(ii,jj);
tokovis(br,4)=tokovip(ii,jj)+sqrt(-1)*tokoviqX(ii,jj)+tokovip(jj,ii)+sq
rt(-1)*tokoviqX(jj,ii);
br=br+1;
end;end;end
%----------------------------------------------------------------------
----
Cvor_i=tokovis(:,1);
Cvor_j=tokovis(:,2);
Pij=real(tokovis(:,3));
Qij=imag(tokovis(:,3));
%----------------------------------------------------------------------
----
pgub_ij=real(tokovis(:,4));
qgub_ij=imag(tokovis(:,4));
pgub_sum=sum(pinj);
qgub_sum=sum(qinj);
qgub_sumX=sum(qgub_ij)/2;
Qshunt=(qgub_sum-qgub_sumX);
%----------------------------------------------------------------------
----
%a+b*P+c*P^2
Fcost=0;
for k=1:ngen
    Cg(k)=a(k)+b(k)*Pg(gcvor(k))+c(k)*Pg(gcvor(k))^2;
    Fcost=Fcost+Cg(k);
```

```
end
%-----------------------------------------------------------------------
----
FdeltaVpq=0;
for k=1:npq
    FdeltaVpq=FdeltaVpq+abs(V(pq(k))-1);
end
%-----------------------------------------------------------------------
----
%=======================================================================
====
lambdaPg=10^6;lambdaVpq=10^6;lambdaQg=10^4;

for k=1:ngen
    if gcvor(k)==sl
if Pg(sl)<Pgmin(k)
    Pgsllim=Pgmin(k);
elseif Pg(sl)>Pgmax(k)
    Pgsllim=Pgmax(k);
else
    Pgsllim=Pg(sl);
end
    end
end
Pf_Pgsl=lambdaPg*(Pg(sl)-Pgsllim)^2;

%Vpqmin=0.95; Vpqmax=1.05 Restricción;
for k=1:npq
    if V(pq(k))<Vpqmin
        Vpqlim(k)=Vpqmin;
    elseif V(pq(k))>Vpqmax
        Vpqlim(k)=Vpqmax;
    else
       Vpqlim(k)=V(pq(k));
    end
    deltaVpq(k)=(V(pq(k))-Vpqlim(k))^2;
end
Pf_Vpq=lambdaVpq*sum(deltaVpq);

%Restricción Reactiva
for k=1:ngen
    if Qg(gcvor(k))<Qgmin(k)
        Qglim(k)=Qgmin(k);
    elseif Qg(gcvor(k))>Qgmax(k)
        Qglim(k)=Qgmax(k);
    else
        Qglim(k)=Qg(gcvor(k));
    end
    deltaQg(k)=(Qg(gcvor(k))-Qglim(k))^2;
end
Pf_Qg=lambdaQg*(sum(deltaQg));

%Funciones Objetivos, por si el cliente quiere agregar nuevas funciones
%=======================================================================
====
switch Case
%       case 'Fcost'              %Fuel cost minimization
%       Fobj=Fcost+Pf_Pgsl+Pf_Vpq+Pf_Qg;
      case 'Ploss'              %Power loss minimization
      Fobj=pgub_sum+Pf_Pgsl+Pf_Vpq+Pf_Qg;
%       case 'VD'               %Voltage profile improvement
%       Fobj=FdeltaVpq+Pf_Pgsl+Pf_Vpq+Pf_Qg;
%       case 'Fcost_Ploss'        %Fuel cost and power loss
minimization
%       Fobj=Fcost+1950*pgub_sum+Pf_Pgsl+Pf_Vpq+Pf_Qg;
%       case 'Fcost_VD'         %Fuel cost minimization and voltage
profile improvement
%       Fobj=Fcost+200*FdeltaVpq+Pf_Pgsl+Pf_Vpq+Pf_Qg;
%       case 'Fcost_Ploss_VD'   %Fuel cost, Ploss minimization and
voltage profile improvement
%       Fobj=Fcost+1950*pgub_sum+200*FdeltaVpq+Pf_Pgsl+Pf_Vpq+Pf_Qg;
end

%-----------------------------------------------------------------------
----
for k=1:npv
Pgener(k)=L(k)*Sbase;
end
tabela_Pg=[pv' Pgener'];
%-----------------------------------------------------------------------
----
for k=(npv+1):(npv+ngen)
Vgen(k-npv)=L(k);
end
tabela_Vg=[gcvor Vgen'];
%-----------------------------------------------------------------------
----
for k=(npv+ngen+1):(npv+ngen+ntf)
ttf(k-npv-ngen)=L(k);
end
tabela_T=[redbrt tfbusi tfbusj ttf'];
%-----------------------------------------------------------------------
----
for k=(npv+ngen+ntf+1):(npv+ngen+ntf+nkom)
Qkond(k-npv-ngen-ntf)=L(k)*Sbase;
end
```

```
tabela_Qc=[kcvor Qkond'];
%-----------------------------------------------------------------------
----
BestResults=[Fobj,Fcost,pgub_sum*Sbase,FdeltaVpq];
minFBEST=Fobj;
maxFBEST=max(FBEST);
meanFBEST=mean(FBEST);
stdFBEST=std(FBEST);
meanTOC=mean(TOC);
Statistics=[BRP,minFBEST,maxFBEST,meanFBEST,stdFBEST,meanTOC];
%-----------------------------------------------------------------------
----
Qc(1:m)=0;Vmin(1:m)=0;Vmax(1:m)=0;Qgmi(1:m)=0;Qgma(1:m)=0;
for k=1:m
    Pg(pq)=0;
    Qg(pq)=0;
    for kk=1:nkom
        if k==kcvor(kk)
            Qc(k)=Qkond(kk);
        end
    end
    if tip(k)==2
        Vmin(k)=Vpqmin;
        Vmax(k)=Vpqmax;
    else
    for kk=1:ngen
        if k==gcvor(kk)
            Vmin(k)=Vgmin(kk);
            Vmax(k)=Vgmax(kk);
            Qgmi(k)=Qgmin(kk);
            Qgma(k)=Qgmax(kk);
        end
    end
    end
end
Qgmin=Qgmi'*Sbase;Qgmax=Qgma'*Sbase;

Vviolating(1:m)=0;Qgviolating(1:m)=0;
for k=1:m
    if V(k)<Vmin(k) || V(k)>Vmax(k)
        Vviolating(k)=k;
    end
    if Qg(k)<Qgmin(k) || Qg(k)>Qgmax(k)
        Qgviolating(k)=k;
    end
end
    Vviolat=find(Vviolating);
    Qgviolat=find(Qgviolating);
```

```
%-----------------------------------------------------------------------
------------------
tabela_cvorovi=[cvor V teta*180/pi Pg*Sbase Qg*Sbase Qc' ppot*Sbase
qpot*Sbase];
tabela_grane=[Cvor_i Cvor_j Pij*Sbase Qij*Sbase pgub_ij*Sbase
qgub_ij*Sbase];
return
```